

(19)日本国特許庁 (J P)

(12)公開特許公報 (A)

(11)特許出願公開番号

特開2000-149053

(P 2 0 0 0 - 1 4 9 0 5 3 A)

(43)公開日 平成12年5月30日(2000.5.30)

(51) Int. Cl. ⁷	識別記号	F I	テーマコード	(参考)
G06T 15/00		G06F 15/72	450	A
1/00		15/66	450	
15/40		15/72	420	

審査請求 未請求 請求項の数16 O L (全18頁)

(21)出願番号	特願平11-225426	(71)出願人	000132471 株式会社セガ・エンタープライゼス 東京都大田区羽田1丁目2番12号
(22)出願日	平成11年8月9日(1999.8.9)	(72)発明者	清水 祐介 東京都大田区羽田1丁目2番12号 株式会社セガ・エンタープライゼス内
(31)優先権主張番号	特願平10-256325	(72)発明者	矢木 博 東京都大田区羽田1丁目2番12号 株式会社セガ・エンタープライゼス内
(32)優先日	平成10年9月10日(1998.9.10)	(74)代理人	100094525 弁理士 土井 健二 (外1名)
(33)優先権主張国	日本 (J P)		

(54)【発明の名称】ブレンディング処理を含む画像処理装置及びその方法

(57)【要約】 (修正有)

【課題】半透明ポリゴンのブレンディング処理を、手前側のポリゴンから順位処理可能にする。

【解決手段】本発明は、半透明ポリゴンのレンダリング処理を、表示画面内の手前側のポリゴンから順に行うこととする特徴とする。そのために、表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと不透明度α_nとが与えられる場合、n枚目のポリゴンの色データT_nの内、当該ポリゴンの不透明度α_n分の成分を、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えて、画像データD_nを求めるレンダリング処理を行う。このレンダリング処理を全てのポリゴンに対して行い、最後の画像データD_nを、表示用の画像データとして利用する。

(1) 本発明

	n	X _n	D _n
初期値	0	1	0
	1	X ₀ (1-α ₁)	D ₀ +T ₁ α ₁ X ₀
	2	X ₁ (1-α ₂)	D ₁ +T ₂ α ₂ X ₁
	3	X ₂ (1-α ₃)	D ₂ +T ₃ α ₃ X ₂
結果	4	X ₃ (1-α ₄)	D ₃ +T ₄ α ₄ X ₃

(2) 従来例

	n	C _n
初期値	-	C ₀
	4	C ₄ = α ₄ T ₄ +(1-α ₄)C ₃
	3	C ₃ = α ₃ T ₃ +(1-α ₃)C ₂
	2	C ₂ = α ₂ T ₂ +(1-α ₂)C ₁
結果	1	C ₁ = α ₁ T ₁ +(1-α ₁)C ₀

【特許請求の範囲】

【請求項1】半透明ポリゴンを含む複数のポリゴンに対する画像処理を行って、表示すべき画像データを生成する画像処理装置において、

表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、不透明度α_nまたは透明度(1-α_n)とが与えられ、

前記手前側から順に前記ポリゴンの色データT_nと不透明度α_nまたは透明度(1-α_n)が供給され、n枚目のポリゴンの色データT_nを、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えるレンダリング処理部を有することを特徴とする画像処理装置。

【請求項2】請求項1において、

前記レンダリング処理部は、n枚目のポリゴンの色データT_nの内、当該ポリゴンの不透明度α_n分の成分を、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えることを特徴とする画像処理装置。

【請求項3】請求項1または2において、

前記レンダリング処理部は、n枚目のポリゴンを重ねた時の透明度の累積された値X_nを逐次求めることを特徴とする画像処理装置。

【請求項4】請求項1または2において、

前記ポリゴンは、前記半透明ポリゴンと不透明ポリゴンとを有し、前記半透明ポリゴンの不透明度α_nは1>α_n>0に、前記不透明ポリゴンの不透明度α_nはα_n=1に、それぞれ設定されることを特徴とする画像処理装置。

【請求項5】請求項1または2において、

前記レンダリング処理部は、n番目のポリゴンに対する処理において、前記透明度の累積値が透明度なしまたは所定の最小値X_{n-1}より小さい場合は、n番目以降のポリゴンに対する処理を省略することを特徴とする画像処理装置。

【請求項6】半透明ポリゴンを含む複数のポリゴンに対する画像処理を行って、表示すべき画像データを生成する画像処理方法において、

表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、不透明度α_nまたは透明度(1-α_n)とが与えられ、

前記手前側から順に前記ポリゴンの色データT_nと不透明度α_nまたは透明度(1-α_n)とを供給する工程と、

n枚目のポリゴンの色データT_nを、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えるレンダリング処理工程を有することを特徴とする画像処理方法。

【請求項7】請求項6において、

前記レンダリング処理工程は、n枚目のポリゴンの色データT_nの内、当該ポリゴンの不透明度α_n分の成分を、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えることを特徴とする画像処理方法。

【請求項8】請求項6または7において、前記ポリゴンは、前記半透明ポリゴンと不透明ポリゴンとを有し、前記半透明ポリゴンの不透明度α_nは1>α_n>0に、前記不透明ポリゴンの不透明度α_nはα_n=1に、それぞれ設定されることを特徴とする画像処理方法。

【請求項9】請求項6または7において、前記レンダリング処理工程は、n番目のポリゴンに対する処理において、前記透明度の累積値が透明度なしまたは所定の最小値X_{n-1}より小さい場合は、n番目以降のポリゴンに対する処理を省略することを特徴とする画像処理方法。

【請求項10】半透明ポリゴンを含む複数のポリゴンに対する画像処理を行って、表示すべき画像データを生成する画像処理手順をコンピュータに実行させるプログラムを記録した記録媒体において、

表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、不透明度α_nまたは透明度(1-α_n)とが与えられ、

前記画像処理手順は、

前記手前側から順に前記ポリゴンの色データT_nと不透明度α_nまたは透明度(1-α_n)とを供給する手順と、

n枚目のポリゴンの色データT_nを、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えて画像データD_nを求めるレンダリング処理手順とを有することを特徴とする画像処理プログラムを記録した記録媒体。

【請求項11】半透明ポリゴンを含む複数のポリゴンに対する画像処理を行って、表示すべき画像データを生成する画像処理方法において、

表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、不透明度α_nまたは透明度(1-α_n)とが与えられ、

前記手前側から順に前記ポリゴンの色データT_nと不透明度α_nまたは透明度(1-α_n)とを供給する工程と、

ワークデータX_nとして、

$$X_n = (1 - \alpha_n) \cdot X_{n-1}$$

を求める、該ワークデータX_nをワークデータ用バッファメモリに記録し、更に画像データD_nとして、

$$D_n = D_{n-1} + T_n \cdot \alpha_n \cdot X_{n-1}$$

を求めるレンダリング処理工程とを有することを特徴とする画像処理方法。

【請求項12】複数のポリゴンに対する所定のレンダリング処理を行って、表示すべき画像データを生成する

画像処理装置において、

表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、ディスティネーション混合係数DE_n及びソース混合係数SR_nについてのデータが与えられ、

前記手前側から順に前記ポリゴンの色データT_nと前記ディスティネーション混合係数DE_n及びソース混合係数SR_nが供給され、n枚目のポリゴンの色データT_nを、前記ディスティネーション混合係数DE_nと、手前に位置するn-1枚のポリゴンのソース混合係数SR₁～SR_{n-1}の累積値X_{n-1}とに応じて、n-1枚目までの画像データD_{n-1}に加えるレンダリング処理部を有することを特徴とする画像処理装置。

【請求項13】請求項12において、

前記レンダリング処理部は、n枚目のポリゴンを重ねた時のソース混合係数SR_nの累積値X_nを逐次求めることを特徴とする画像処理装置。

【請求項14】請求項12または13において、

前記レンダリング処理部は、n番目のポリゴンに対する処理において、前記ソース混合係数SR_nの累積値が所定の最小値X_{n-1}より小さい場合は、n番目以降のポリゴンに対する処理を省略することを特徴とする画像処理装置。

【請求項15】複数のポリゴンに対する所定のブレンディング処理を行って、表示すべき画像データを生成する画像処理方法において、

表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、ディスティネーション混合係数DE_n及びソース混合係数SR_nについてのデータが与えられ、

前記手前側から順に前記ポリゴンの色データT_nと前記ディスティネーション混合係数DE_n及びソース混合係数SR_nを供給する工程と、

n枚目のポリゴンの色データT_nを、前記ディスティネーション混合係数DE_nと、手前に位置するn-1枚のポリゴンのソース混合係数SR₁～SR_{n-1}の累積値X_{n-1}とに応じて、n-1枚目までの画像データD_{n-1}に加えるレンダリング処理工程とを有することを特徴とする画像処理装置。

【請求項16】複数のポリゴンに対する所定のブレンディング処理を行って、表示すべき画像データを生成する画像処理手順をコンピュータに実行させるプログラムを記録した記録媒体において、

表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、ディスティネーション混合係数DE_n及びソース混合係数SR_nについてのデータが与えられ、

前記画像処理手順は、

前記手前側から順に前記ポリゴンの色データT_nと前記ディスティネーション混合係数DE_n及びソース混合係数

SR_nを供給する手順と、

n枚目のポリゴンの色データT_nを、前記ディスティネーション混合係数DE_nと、手前に位置するn-1枚のポリゴンのソース混合係数SR₁～SR_{n-1}の累積値X_{n-1}とに応じて、n-1枚目までの画像データD_{n-1}に加えるレンダリング処理手順とを有することを特徴とする画像処理プログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

10 【発明の属する技術分野】本発明は、半透明ポリゴンを含む複数のポリゴンに対する画像処理装置及びその方法にかかり、特に、半透明ポリゴンの場合でも表示画面内の手前から順にポリゴンデータを画像データに変換処理することができる画像処理装置及びその方法に関する。更に、本発明は、複数のポリゴンに対するブレンディング処理を表示画面内の手前から順にポリゴンデータに対して行うことができる画像処理装置及びその方法に関する。

【0002】

20 【従来の技術】ゲーム装置やシミュレーション装置において、操作入力に応じてキャラクタ等を構成する複数のポリゴンの位置を求め、それらのポリゴンのうち表示されるポリゴンに対してレンダリング処理を行って画像データを生成し、それらを表示することが行われる。

【0003】半透明ポリゴンに対するレンダリング処理は、不透明ポリゴンと異なり、その後方（奥側）に位置する他のポリゴンとのブレンディング処理を伴い、従来は、表示画面内の奥側の半透明ポリゴンから順にブレンディング処理を含むレンダリング処理を行うことが必要である。

【0004】一方、不透明のポリゴンに対するレンダリング処理は、表示画面内の奥行きを示すZ値を利用した陰面消去処理を行うことで、複数のポリゴンのレンダリングの順番に制約を課すことなく、重なり合う部分の処理が行われる。また、不透明ポリゴンと半透明ポリゴンとが混在する場合は、上記のZ値を利用した陰面消去処理によって不透明ポリゴンを任意の順番で処理し、その後、半透明ポリゴンを奥側から順に処理する方法が一般的である。

【0005】

【発明が解決しようとする課題】ところが、レンダリング処理において、処理すべきポリゴン数が多くなると、決められたフレーム間の時間内でレンダリング処理が完了しない場合がある。その場合、画像処理の質を重視すると全ての半透明ポリゴンのレンダリング処理が終了するまで待たなければならず、表示中のキャラクタの動きが停止することになり、ゲーム装置あるいはシミュレーション装置において不都合である。また、キャラクタの動きを重視するとフレーム間の時間内で処理されないポリゴンが存在することになる。その場合、半透明ポリゴ

ンのレンダリング処理が、より奥側のポリゴンから処理されるので、画像内で最も重要なキャラクタの一つである手前側のポリゴンの処理が切り捨てられることになる。

【0006】更に、不透明ポリゴンと半透明ポリゴンのレンダリング処理は、上記した通り、それぞれの効率を重視すると、それらのポリゴンを分けて別々に処理する必要があり、画像処理の負担を増大させる。かかる負担の増大は、上記と同様にフレーム間の時間内での画像処理の完了を困難にさせる。

【0007】更に、半透明ポリゴンに対する不透明度を利用したブレンディング処理に限定されず、一般的なブレンディング処理においても、表示画面のより奥側のポリゴンの色データを先に求め、その色データと次の処理するポリゴンの色データに対して所定のソース混合計数とディスティネーション混合計数によりブレンディング処理することが提案されている。この場合も、表示画面の奥側のポリゴンから順番に処理することが要求され、半透明ポリゴンの場合と同様の課題を有する。

【0008】そこで、本発明の目的は、半透明ポリゴンの画像処理において、フレーム間の時間内により手前側のポリゴンのレンダリング処理を確実に行うことができる画像処理装置、その方法及びそれをコンピュータに実行させるプログラムを記録した記録媒体を提供することにある。

【0009】更に、本発明の目的は、不透明ポリゴンと半透明ポリゴンが混在したポリゴンの画像処理において、両ポリゴンを区別することなくレンダリング処理することができ、更に処理効率を高めた画像処理装置、その方法及びそれをコンピュータに実行させるプログラムを記録した記録媒体を提供することにある。

【0010】更に、本発明の目的は、一般的なブレンディング処理において、処理効率を高めた画像処理装置、その画像処理方法及びそれをコンピュータに実行させるプログラムを記録した記録媒体を提供することにある。

【0011】

【課題を解決するための手段】上記の目的を達成する為了、本発明は、半透明ポリゴンのレンダリング処理を、表示画面内の手前側のポリゴンから順に行うことの特徴とする。そのために、表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと不透明度α_nとが与えられる場合、ワークデータX_nとして、

$$X_n = (1 - \alpha_n) \cdot X_{n-1}$$

を求め、更に画像データD_nとして、

$$D_n = D_{n-1} + T_n \cdot \alpha_n \cdot X_{n-1}$$

を求め、ワークデータX_n及び画像データD_nを求める上記の演算処理を全てのポリゴンに対して行う。そして、最後の画像データD_nを、表示用の画像データとして利用する。上記において、ワークデータX_nの初期値X₀は、例えば1が好ましい。また、画像データD_nの

初期値D₀は、例えば0が好ましい。

【0012】上記の画像処理によれば、半透明ポリゴンを手前側から順に処理することができる、フレーム間の時間が短い場合でも表示画面内の重要度の高いより手前側に位置するポリゴンのレンダリング処理が省略されることがない。従って、ポリゴンで構成される重要なキャラクタの動きを止めることなく、スムーズな動きの画像を再現することができる。

【0013】上記のワークデータX_nは、n枚の半透明

10 ポリゴンの透明度(1-α)が累積された値を意味する。そして、複数のポリゴンが重なり合う場合は、手前側のポリゴンから順番にレンダリング処理し、それぞれのポリゴンのレンダリング処理では、奥側のポリゴンの色データT_nを、それより手前のn-1枚のポリゴンの透明度を累積した値X_{n-1}に応じて、手前のn-1枚のポリゴンが重ねられた画像データD_{n-1}に加える。

【0014】従って、本発明では、手前側のポリゴンから順番に上記のレンダリング処理しながら、重ね合わせたポリゴンの透明度の累積された値を、各ピクセル毎に20 求めて記録しておくことが必要である。

【0015】更に、それぞれのポリゴンのレンダリング処理では、処理中のポリゴンの透明度に応じて、その色データT_nを加えることが望ましい。即ち、処理中のポリゴンの不透明度α_nに応じた色データT_nの成分を、それより手前のn-1枚のポリゴンの透明度(1-α)の累積された値X_{n-1}に応じて、n-1枚重なったポリゴンの画像データD_{n-1}に加えることが望ましい。

【0016】上記の目的を達成する為に、本発明は、半透明ポリゴンを含む複数のポリゴンに対する画像処理を行って、表示すべき画像データを生成する画像処理装置において、表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データT_nと、不透明度α_nまたは透明度(1-α_n)とが与えられ、前記手前側から順に前記ポリゴンの色データT_nと不透明度α_nまたは透明度(1-α_n)が供給され、n枚目のポリゴンの色データT_nを、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えるレンダリング処理部を有することを特徴とする。

40 【0017】更に、好ましくは、上記の発明において、前記レンダリング処理部は、n枚目のポリゴンの色データT_nの内、当該ポリゴンの不透明度α_nの成分を、手前に位置するn-1枚のポリゴンの透明度の累積値X_{n-1}に応じて、n-1枚目までの画像データD_{n-1}に加えることを特徴とする。

【0018】上記の発明によれば、複数の半透明ポリゴンを手前から順番にレンダリング処理することができる。従って、フレーム間の時間内に全てのポリゴンの処理を終了しなくても、画像データの品質を損なうことは50 ない。

【0019】更に、上記の発明において、前記ポリゴンは、前記半透明ポリゴンと不透明ポリゴンとを有し、前記半透明ポリゴンの不透明度 α_1 は $1 > \alpha_1 > 0$ に、前記不透明ポリゴンの不透明度 α_2 は $\alpha_2 = 1$ に、それぞれ設定されることを特徴とする。この発明によれば、不透明ポリゴンと半透明ポリゴンとを区別することなく、手前から順番に一緒にレンダリング処理することができる。

【0020】更に、上記の発明において、前記レンダリング処理部は、n番目のポリゴンに対する処理において、前記透明度の累積値が透明度なしまたは所定の最小値 X_{n-1} 。より小さい場合は、n番目以降のポリゴンに対する処理を省略することを特徴とする。上記の構成により、不透明ポリゴンの背面側（より奥側）に重なるポリゴンの処理や、不透明度 α の高い半透明ポリゴンの背面側（より奥側）に重なるポリゴンの処理を省略することができ、レンダリング処理の効率を上げることができる。

【0021】更に、上記の目的を達成するために、本発明の別の側面では、一般的なブレンディング処理に対しても表示画面の手前側のポリゴンから順に処理する。この発明は、複数のポリゴンに対する所定のブレンディング処理を行って、表示すべき画像データを生成する画像処理装置において、表示画面内の手前側から順に並べた時のn番目のポリゴンに、色データ T_n と、ディスティネーション混合係数 DE_n 及びソース混合係数 SR_n についてのデータが与えられ、前記手前側から順に前記ポリゴンの色データ T_n と前記ディスティネーション混合係数 DE_n 及びソース混合係数 SR_n が供給され、n枚目のポリゴンの色データ T_n を、前記ディスティネーション混合係数 DE_n と、手前に位置する $n-1$ 枚のポリゴンのソース混合係数 $SR_{n-1} \sim SR_1$ の累積値 X_{n-1} とに応じて、 $n-1$ 枚目までの画像データ D_{n-1} に加えるレンダリング処理部を有することを特徴とする。

【0022】上記の発明において、ディスティネーション混合係数 DE_n 及びソース混合係数 SR_n を適宜選択することにより、様々なブレンディング処理を行うことができ、汎用的なブレンディング処理を行うレンダリング処理部を提供することができる。

【0023】更に、本発明は、上記の画像処理方法、その画像処理方法をコンピュータに実行させるプログラムを記録した記録媒体を提供する。

【0024】

【発明の実施の形態】以下、本発明の実施の形態について図面に従って説明する。しかしながら、本発明の技術的範囲がその実施の形態に限定されるものではない。

【0025】図1は、半透明ポリゴンの重なり例を説明

$$C_1 = T_1 \alpha_1 + T_2 \alpha_2 (1 - \alpha_1) + C_4 (1 - \alpha_1) (1 - \alpha_2) \quad (5)$$

ここで、ポリゴン P_4 が不透明ポリゴンであり、不透明度データ $\alpha_4 = 1$ とし、更に、任意のワークデータ X_0

する図である。この例では、3枚の半透明ポリゴン P_1 、 P_2 、 P_3 と、1枚の不透明ポリゴン P_4 とが、表示画面内の奥行き方向（Z値）について手前から順に重なっている。従って、領域 C_4 の画像データ C_4 は、不透明ポリゴン P_4 のテクスチャデータから環境光等に応じて形成される色データ T_4 、そのままである。

【0026】本明細書において、ポリゴンの色データ T は、例えばポリゴン自体の色データ、模様を含む色データであるテクスチャデータ、更に、テクスチャデータやポリゴン自体の色データにシェーディング処理した後の色データ等を含む。それらのいずれかの色データを代表して、ポリゴンの色データ T と称する。また、ブレンディング処理などにより求められた色データは、表示に利用される画像データ C である。従って、画像データ C も一種の色データである。

【0027】

$$\text{即ち}, C_1 = T_1 \quad (1)$$

また、領域 C_3 は、不透明ポリゴン P_4 上に半透明ポリゴン P_3 が重なっているので、その画像データは、半透明ポリゴン P_3 の不透明度 α_3 ($\alpha_3 = 1$ で不透明、 $\alpha_3 = 0$ で透明、 $1 > \alpha_3 > 0$ で半透明) 及び色データ T_3 に従って、ポリゴン P_4 、 P_3 の色データをブレンディングした色データとなる。即ち、従来の一般的なレンダリング方法によれば、領域 C_3 の色データは、

$$C_3 = \alpha_3 T_3 + (1 - \alpha_3) C_4 \quad (2)$$

で求められる。

【0028】同様に、領域 C_2 の画像データ C_2 も、半透明ポリゴン P_2 の不透明度 α_2 及び色データ T_2 と領域 C_3 の色データ C_3 とから、

$$C_2 = \alpha_2 T_2 + (1 - \alpha_2) C_3 \quad (3)$$

で求められ、更に、領域 C_1 の画像データ C_1 も、

$$C_1 = \alpha_1 T_1 + (1 - \alpha_1) C_2 \quad (4)$$

で求められる。

【0029】以上の通り、半透明ポリゴンに対するレンダリング処理は、他のポリゴンとの重なり合う領域において、より奥側の画像データ C_{n-1} を再帰的に参照してブレンディング処理を行う必要があり、表示画面内の奥側から順にポリゴンのレンダリング処理を行うのが一般的である。

【0030】そこで、本発明は、上記した従来の課題を解決するために、表示画面内の手前側から順にポリゴンのレンダリング処理を行うことを特徴とする。本発明によれば、上記の式 (2)、(3)、(4) から、 C_1 について開くと、

$$C_1 = \alpha_1 T_1 + (1 - \alpha_1) \{ \alpha_2 T_2 + (1 - \alpha_2) [\alpha_3 T_3 + (1 - \alpha_3) C_4] \}$$

となり、更に、 T_n についてまとめると、

$$C_1 = T_1 \alpha_1 + T_2 \alpha_2 (1 - \alpha_1) + T_3 \alpha_3 (1 - \alpha_2) (1 - \alpha_3) \quad (5)$$

ここで、ポリゴン P_4 が不透明ポリゴンであり、不透明度データ $\alpha_4 = 1$ とし、更に、任意のワークデータ X_0

として、

$$\begin{aligned} X_n &= (1 - \alpha_1) (1 - \alpha_2) \dots (1 - \alpha_n) \\ &= X_{n-1} (1 - \alpha_n) \end{aligned} \quad (6)$$

と定義するとする。但し、 $X_0 = 1$ とする。

$$\begin{aligned} C_1 &= T_1 \alpha_1 X_0 + T_2 \alpha_2 X_1 + T_3 \alpha_3 X_2 + C_4 X_3 \\ &= T_1 \alpha_1 X_0 + T_2 \alpha_2 X_1 + T_3 \alpha_3 X_2 + T_4 \alpha_4 X_3, \end{aligned}$$

となり、各々の T_i, α_i, X_{i-1} に着目すれば、

$$D_n = D_{n-1} + T_n \alpha_n X_{n-1} \quad (7)$$

とすることができる。但し、 $D_0 = 0$ である。その結果、 $n = 4$ の場合は、

$$D_4 = C_4$$

が成立する。

【0032】即ち、従来のブレンディング処理と同じ処理を、上記の式(6)と(7)を利用してすることで、表示画面内の手前側から順にポリゴンのデータとして、色データ T_i とその不透明度 α_i を与えて、全ての重なり合うポリゴンに対する演算処理を行うことにより、最終的に得られる画像データ D_n は、図1の領域C1の画像データ C_1 と同じになる。従って、新たに定義したワークデータ X_n を利用して、画像データ $D_n = D_{n-1} + T_n \alpha_n X_{n-1}$ を求めることができる。これらの数式のインデックスから明らかな通り、これらの演算において、より奥側のポリゴンのデータを再帰的に呼び出す必要はなくなる。

【0033】図2は、上記の演算を利用する本発明のレンダリング処理を、図1の例に適用した時の演算式を示す図表である。図2(1)に、本発明の場合を、図2(2)に従来例の場合をそれぞれ示す。

【0034】図2の図表から明らかな通り、従来例のレンダリング処理によれば、半透明のポリゴンに対して、奥側のポリゴン P_1 の色データ T_1 と不透明度 α_1 と初期値 C_1 を利用して、領域C4の画像データ C_4 が求められ、更に、一つ手前のポリゴン P_2 の色データ T_2 と不透明度 α_2 と初期値 C_1 を利用して、領域C3の画像データ C_3 が求められる。更に、領域C2の画像データ C_2 及び領域C1の画像データ C_1 が順に求められる。

【0035】それに対して、本発明によれば、上記の式(6)、(7)を利用してすることで、最初に最も手前に位置するポリゴン P_1 の色データ T_1 と不透明度 α_1 と初期値 X_0 と D_0 を利用して、途中の画像データ D_1 が求められ、更に、一つ奥側のポリゴン P_2 の色データ T_2 と不透明度 α_2 と上記の途中の画像データ D_1 を利用して、次の途中の画像データ D_2 が求められる。更に、同様にして次の途中の画像データ D_3, D_4 が求められる。そして、全てのポリゴンのデータに対する処理が終了した最終画像データ D_4 が、領域C1の画像データ C_1 として利用される。

【0036】従って、本発明によれば、半透明ポリゴンに対するレンダリング処理は、手前側のポリゴンに対するデータ(色データ T と不透明度データ α)を利用して

【0031】上記式(6)を式(5)に挿入すると、

$$= T_1 \alpha_1 X_0 + T_2 \alpha_2 X_1 + T_3 \alpha_3 X_2 + C_4 X_3$$

(7)

次々に演算処理することができる。更に、後述する具体

10 例で明らかになるが、より奥側のポリゴンに対するレンダリング処理では、ワークデータ X_{n-1} が小さくなる傾向にあり、そのワークデータ X_{n-1} を利用して求められる途中の画像データ D_n も変化が少なくなる傾向にある。従って、ある程度ワークデータ X_{n-1} が小さくなると、あるいは0になると、それ以降の奥側のポリゴンのデータに対するレンダリング処理を省略しても、表示される画像への影響は殆どないことが理解される。

【0037】更に、上記のレンダリング処理の演算において、不透明ポリゴンに対しては、単に不透明度 α を

20 1(不透明) とすれば、上記の演算をそのまま適用して処理することができる。即ち、不透明度 $\alpha = 1$ の場合は、ワークデータ $X_n = 0$ となり、それより奥側のポリゴンのレンダリング処理の結果に変化はなくなり、実質的にそれらの奥側のポリゴンに対するレンダリングは不要になる。従って、本発明のレンダリング処理を利用することにより、半透明ポリゴンと不透明ポリゴンとを混在させても、より手前側のポリゴンから順に、半透明と不透明を区別なく処理することができる。

【0038】上記したワークデータ X_n は、物理的には、 n 枚のポリゴンを重ねた時の透明度 $(1 - \alpha)$ の累積された値を意味する。即ち、上記の式(6)から明らかな通り、ワークデータ X_n は、各ポリゴンの透明度

$(1 - \alpha)$ が全て積算された値である。つまり、透明度の高いポリゴン、例えば α が 1 に近いポリゴンが重ねられる場合は、その奥側に位置する n 枚目のポリゴンの色データ T_n を追加する場合は、それより手前の $n - 1$ 枚の重なったポリゴンの透明度の累積値に応じて、画像データ D_{n-1} に影響を与える。そして、望ましくは、 n 枚目のポリゴンの不透明度 α_n に応じた色データ T_n が、

40 加えられる。 n 枚目のポリゴン自身が透明なポリゴン $(\alpha_n = 1)$ の場合は、その色データ T_n は加えられるべきではないからである。

【0039】以上のように、上記の式(7)

$$D_n = D_{n-1} + T_n \alpha_n X_{n-1}$$

によるレンダリング処理は、第1に、処理しようとする奥側のポリゴンの色データ T_n を、それより手前の $n - 1$ 枚のポリゴンの透明度を累積した値 X_{n-1} に応じて、手前の $n - 1$ 枚のポリゴンが重ねられた画像データ D_{n-1} に加える処理である。

50 【0040】第2に、上記の式(7)によるレンダリン

グ処理は、処理中のポリゴンの透明度に応じて、その色データ T_n を加えることを行う。即ち、処理中のポリゴンの不透明度 α_n に応じた色データ T_n の成分を、それより手前の $n-1$ 枚のポリゴンの透明度の累積された値 X_{n-1} に応じて、 $n-1$ 枚重なったポリゴンの画像データ D_{n-1} に加えることを意味する。

【0041】そして、透明度の累積値 X_n の初期値 $X_0 = 1$ の意味は、最初のポリゴンより手前には、別のポリゴンが存在しないで、透明度は最大であることを意味する。従って、最初のポリゴンの色データ T_1 は、そのまま画像データ D_1 として利用される。但し、最初のポ

$$T_1 = \text{赤} \cdots R = 1.0, \quad G = 0,$$

$$T_2 = \text{青} \cdots R = 0, \quad G = 0,$$

$$T_3 = \text{黄色} \cdots R = 1.0, \quad G = 1.0,$$

$$T_4 = \text{黒色} \cdots R = 0, \quad G = 0,$$

の場合である。これらのポリゴンデータがそれぞれの左側の図表に順番が逆に示され、それらのポリゴンデータを利用したレンダリング結果が、右側の図表に示される。

【0043】このレンダリング結果に示される通り、最終的に得られた画像データ D_4 は、従来のレンダリング演算により求めた画像データ C_1 と同じになっていることが理解される。

【0044】図3に示した本発明のレンダリング処理を、物理的な意味で解説すると、1枚目のポリゴン ($n = 1$) については、その不透明度 α_1 に応じた色データ T_1 の成分が、そのまま画像データ D_1 となる。次に、2枚目のポリゴン ($n = 2$) については、その不透明度 α_2 に応じた色データ T_2 の成分が、それより手前の1枚目のポリゴンの透明度 ($1 - \alpha_1$) に応じて、画像データ D_1 に加えられる。つまり、2枚目のポリゴンの青Bの色データ 1.000 のうち、それ自身の不透明度 $\alpha_2 = 0.5$ に応じた成分 0.5 が、それより手前の1枚目のポリゴンの透明度 $1 - \alpha_1 = 0.5$ に応じて、1枚目までの青Bのデータ 0.0 に加えられて、 0.25 となる。

【0045】更に、3枚目のポリゴンの赤Rと緑Gの色データ 1.0 は、その透明度 $\alpha_3 = 0.5$ 分の成分が、1枚目と2枚目のポリゴンの透明度の累積値 $X_2 = 0.5 \times 0.5 = 0.25$ に応じて、2枚目までの赤Rと緑Gの画像データ、 0.5 及び 0.0 にそれぞれ加えられる。従って、それぞれ 0.125 だけ加えられる。最後に、4枚目のポリゴンは、不透明度 $\alpha_4 = 1$ であるが、色データ T_4 が全て 0 であるので、最終的な画像データに変化はない。

【0046】以上の通り、図3の例のレンダリング処理は、手前のポリゴンから順に処理し、次の奥側のポリゴンに対するレンダリング処理は、処理中のポリゴンの色データの不透明度に応じた成分を、それより手前の重なり合ったポリゴンの透明度の累積された値に応じて、画

リゴン自身の不透明度 α_n に応じた色データ T_n の成分が画像データ D_n として利用される。尚、画像データ D_n の初期値 $D_n = 0$ の意味は、輝度値が 0 であることを意味し、何ら色の成分がない真っ黒の状態を意味する。画像データの初期値 D_n は、必ずしも真っ黒である必要はない。

【0042】図3は、図2に示したレンダリング処理の演算を具体例に適用した例を示す図表である。ここでの具体例は、図1の各ポリゴン $P_1 \sim P_4$ の色データ（テクスチャ、環境光、等を考慮して得られる色データ） $T_1 \sim T_4$ 及び不透明度 $\alpha_1 \sim \alpha_4$ が、

$$T_1 = \text{赤} \cdots R = 1.0, \quad G = 0, \quad B = 0, \quad \alpha_1 = 0.5$$

$$T_2 = \text{青} \cdots R = 0, \quad G = 0, \quad B = 1.0, \quad \alpha_2 = 0.5$$

$$T_3 = \text{黄色} \cdots R = 1.0, \quad G = 1.0, \quad B = 0, \quad \alpha_3 = 0.5$$

$$T_4 = \text{黒色} \cdots R = 0, \quad G = 0, \quad B = 0, \quad \alpha_4 = 1.0$$

像データに加える処理である。

【0047】図3に示した本発明のレンダリング結果の図表から理解される通り、手前側のポリゴンから順にレンダリング演算が行われる場合、そのワークデータ X_n は、徐々に小さくなる。そして、 $X_4 = 0$ になると、図示されないが、その後の演算によっても画像データ D_n は変化しない。また、図示されないが、ワークデータ X_{n-1} が小さくなると、次に求められる画像データ D_n は、

$$D_n = D_{n-1} + T_n \alpha_n X_{n-1}$$

の式から明らかな通り、それほど前の画像データ D_{n-1} と変わらない。

【0048】従って、フレーム間の時間内の限られた演算時間において、後回しにされているより奥側のポリゴンに対するレンダリング処理がカット、省略されても、表示される画像データにそれほど大きな影響を与えることはない。

【0049】上記にかかる物理的な意味を説明すると、重なり合うポリゴンの透明度の累積値であるワークデータ X_{n-1} が小さくなるとは、透明度が悪くなることを意味し、その場合は、それより奥側のポリゴンは、画像データにほとんど影響を与えないことを意味する。

【0050】図4は、本発明の画像処理装置にかかるゲーム装置の概略構成例を示す図である。図4に示されたゲーム装置では、CPU20がゲームプログラムを実行し、ゲームの進行に同期して変化する画像を生成するための描画コマンドを発生する。この描画コマンドには、例えば、ポリゴンの頂点座標とその頂点のテクスチャ座標や不透明度データが含まれる。また、CPU20は、ゲームプログラムの実行時に使用するためのワークRAM22とバス24を介して接続される。

【0051】バス24は、CPU20ともっぱら画像の描画を行うレンダラ34との間のインターフェースとなるシステムコントローラ26に接続される。システムコントローラ26は、CPU20により発行される描画コ

マンドや、その他のコマンドをレンダラ34に転送する。また、システムコントローラ26は、外部インターフェース28に接続され、ゲームプログラムやゲームに必要なオブジェクトのデータが記録されたCD-ROM等の様な外部記憶媒体30と、操作入力を入力するためのコントロールパッド32に接続される。

【0052】画像の描画を行うレンダラ34は、例えビデオデジタルプロセッサで構成され、バス38を介してフレームバッファ40と、Zデータバッファ42と、テクスチャバッファメモリ44と、Xバッファメモリ45とに接続される。レンダラ34は、システムコントローラ26を経由してCPU20から送られる描画コマンド及びその他のコマンドに応答して、テクスチャデータをテクスチャバッファ44から読み出され、また、レンダリング処理を行って、描画すべき画像データをフレームバッファ40に格納する。また、レンダラ34は、その描画されるピクセルの表示画面内の奥行きを示すZデータをZデータバッファ42内の対応する位置に格納する。そして、1フレーム分の画像データをフレームバッファ40内に格納し終わると、レンダラ34は、その画像データを図示しないモニタに転送しモニタ画面にゲームの画像を表示させる。

【0053】ゲーム装置のプレーヤによって入力される操作入力は、コントロールパッド32からシステムコントローラ26を経由してワークRAM22に格納される。この操作入力に応答して、CPU20はゲームプログラムを実行し、リアルタイムに生成されるオブジェクトの描画コマンドを発行する。この描画コマンドには上記の通りポリゴンデータが含まれるが、事前に奥行きを示すZデータに従ってソート処理を行うことにより、描画コマンドは表示画面の手前側のポリゴンから順に、レンダラ34に供給される。

【0054】図5は、本発明の実施の形態例におけるレンダラの構成図である。図5には、図4で示したZ値バッファ42とテクスチャバッファ44と共に、レンダラ34内部の一部の構成が示される。即ち、制御ブロック341は、システムコントローラ26とバス36を介して接続され、更にバス38を介してテクスチャバッファ44、Z値バッファ42と接続される。制御ブロック341は、CPU20が発行する描画コマンドを受信し、テクスチャバッファ44からテクスチャデータ（一種の色データ）を読み出し、またZ値バッファ内のZ値に従って所定の処理を行い、更に、レンダラ34内のレンダリング処理演算を制御する。図5の実施の形態例では、ポリゴンの色データとしてテクスチャデータTは、テクスチャデータTと置き換えられる。

【0055】制御ブロック341は、更に、ワークデータX_oを所定のピクセル毎に格納するXバッファメモリ342と、画像データD_oを所定のピクセル毎に格納す

るDバッファメモリ343とも接続され、それらのバッファメモリ342、343に初期値を与えると共に、演算結果をそれぞれ格納する。尚、Xバッファメモリ342、Dバッファメモリ343は、図4のXバッファ45、フレームバッファ40とそれぞれ同等であり、それに代わるバッファメモリである。

【0056】乗算器344は、Xバッファ342からのワークデータX_oと描画コマンドと共に供給された処理中のポリゴン内の処理ピクセルの不透明度 α_o とから、それらの乗算値 $\alpha_o X_{o-1}$ を生成する。また、減算器345は、その乗算値 $\alpha_o X_{o-1}$ とワークデータX_{o-1}とから、それらの減算値 $X_{o-1} (1 - \alpha_o)$ を生成し、Xバッファ342内のものとのピクセルに対応する領域に、新たなワークデータX_oとして、減算値 $X_{o-1} (1 - \alpha_o)$ を記録する。

【0057】一方、乗算器346は、乗算器344が生成した乗算値 $\alpha_o X_{o-1}$ と処理中のポリゴン内の処理ピクセルに対応するテクスチャデータ（色データ）T_oとの乗算値T_o $\alpha_o X_{o-1}$ を生成する。また、加算器347は、その乗算値T_o $\alpha_o X_{o-1}$ とDバッファ343内の画像データD_{o-1}との加算値D_{o-1} + T_o $\alpha_o X_{o-1}$ を生成し、処理済の画像データD_oとして、Dバッファメモリ343の対応するピクセルの領域内に記憶される。

【0058】制御ブロック341は、フレーム間の時間内で上記のレンダリング処理を完了し、最後に累積されたDバッファメモリ343内の画像データDを、図示しない表示装置に供給し、画像を表示させる。また、制御ブロック341は、Xバッファ342内のワークデータX_oの値を監視し、ワークデータX_{o-1}が所定の最小値X_{min}より小さい、あるいはワークデータX_{o-1}が0の場合は、それ以降のレンダリング処理演算を中止すべく、Xバッファ342及びDバッファ343からのデータX_{o-1}とD_{o-1}の出力を行わず、また、描画コマンドから得られるあるいは生成されるテクスチャデータT_o及び不透明度データ α_o の供給を行わない。この様に、制御ブロック341は、レンダリング処理演算の開始と終了を制御する。

【0059】図6は、本実施の形態例の画像処理のフローチャート図である。図6に従って、1つのフレームに対する画像処理の流れを説明する。まず、コントロールパッド32から供給される操作入力に応答して、CPU20がRAM22内のゲームプログラムを実行し、表示すべきポリゴンデータを生成する。このポリゴンデータは、表示画面内の奥行き方向の手前から順にソーティングされる(S10)。このポリゴンには、半透明ポリゴンと不透明ポリゴンが含まれる。ポリゴンデータは、例えば、頂点座標とそれら頂点の不透明度データ、色データ、テクスチャ座標データ等が含まれる。

【0060】ポリゴンのテクスチャデータは、予めテク

スチャバッファ44内に記録される。そして、描画コマンドに含まれるポリゴンのテクスチャ座標から処理対象のピクセルのテクスチャ座標が求められ、テクスチャバッファ44内のテクスチャデータが読み出される。このテクスチャデータと、上記のポリゴンの属性データである白色光での色データ及び環境光の色データ等から、ポリゴンのピクセルの色データが求められる。この色データを、本明細書では総称してテクスチャデータと称する。

【0061】レンダラ34内の制御ブロックは、最初にレンダラ内のXバッファ342とDバッファ343とに初期値 $X_0 = 1, D_0 = 0$ を記録する(S12)。そして、手前から順に描画コマンドがポリゴンデータと共に発行される(S14)。このポリゴンデータから、制御ブロック341は、テクスチャバッファ44を参照して、処理中のピクセルに対するテクスチャデータ(色データ) T_0 を生成する。また、ポリゴンデータから、処理中のピクセルに対する不透明度データ α_0 を生成する。

【0062】そして、図5に示した乗算器344, 346, 減算器345及び加算機346を利用して、ワークデータ X_0 と途中の画像データ D_0 とを求め、それらのデータ X_0, D_0 を新たにXバッファ342及びDバッファ343に記録する(S18)。即ち、工程S18のレンダリング処理は、n枚目のポリゴンを重ねた時の透明度の累積された値 X_n を求めると共に、n枚目のポリゴンのテクスチャデータ(色データ) T_n のうち、n枚目のポリゴンの不透明度 α_n に応じた成分が、手前に位置する $n-1$ 枚のポリゴンの透明度の累積値 X_{n-1} に応じて、 $n-1$ 枚目までの画像データ D_{n-1} に加えられる処理である。

【0063】上記のピクセルのワークデータ X 、画像データ D の生成は、ポリゴン内の全てのピクセルについて繰り返し行われる(S20)。但し、レンダリング演算の結果得られたワークデータ X_{n-1} が0になるか、或いはワークデータ X_{n-1} が予め設定した最小値 X_{min} 未満になると、それ以降のより奥側のポリゴンに対するピクセルのレンダリング演算の処理は省略され、次のピクセルのレンダリング処理が行われる(S16)。

【0064】1つのポリゴン内のピクセルについてのレンダリング処理が終了すると、次のより奥側のポリゴンに対するレンダリング処理が引き続き行われる(S22)。そして、最後にDバッファ343内に記録された画像データ D が、表示装置に供給され、その画像が表示される。

【0065】上記のフレームの画像データの生成において、フレーム間の時間を経過する場合は、それ以降のよ

$$F B_m = P I X_m * D E_m + F B_{m-1} * S R_m$$

となる。但し、mは整数であり、小さいほど先に処理される(*は積を表す。以下同様)。

り奥側のポリゴンに対するレンダリング処理を中止することができる。即ち、制御ブロック341が、フレーム間の時間を管理し、レンダリングすべきポリゴンの数が多く、時間内に処理完了しない場合は、そのフレームのレンダリング処理を強制的に終了することができる。そのような強制的な終了をしても、表示される画像内の比較的重要な手前に位置するポリゴンの処理は完了しているので、画像のスムーズな動きを乱すことはない。

【0066】尚、ポリゴンの形状、位置によっては、2つのポリゴンが奥行き方向で交差する場合もある。かかる場合は、一方のポリゴンの一部が他方のポリゴンの一部よりも手前側に位置し、一方のポリゴンの残りの部分が他方のポリゴンの残りの部分より奥側に位置する。そのような場合でも、上記のレンダリング処理では、より手前側のピクセルを最初に処理することが必要になる。従って、その場合は、処理中のピクセルのZ値とZ値バッファ内の記録済のピクセルのZ値とを比較し、処理中のピクセルのZ値がより奥側に位置する場合は、上記のレンダリング演算処理を行うようになることが有効である。

【0067】上記の実施の形態例では、フローチャートにおいて、不透明ポリゴンと半透明ポリゴンとが混在する場合も、一緒に処理することができるることを説明したが、従来の通り、不透明ポリゴンと半透明ポリゴンとを区別して処理する場合に、半透明ポリゴンの処理に上記のレンダリング処理を利用してもよい。その場合でも、一定の時間内に処理できるポリゴン数が処理すべきポリゴン数よりも少ない場合でも、再生される画像に大きな影響を与えることなく、画像のスムーズな動きを維持して再生することができる。

【0068】〔汎用的なブレンド処理への適用〕上記の実施の形態例では、半透明ポリゴンの色とその奥側に位置するポリゴンの色とを不透明度 α で混合ブレンディング処理する場合に、手前側のポリゴンから順番に処理する例を説明した。しかしながら、一般的なコンピュータを利用したグラフィックライブラリには、例えばオープンGL (Graphic Library) で規格化されているように、多用なブレンディング処理ができるようになっている。その一般化されたブレンディング処理は、以前処理して求めたフレームバッファ内の色データ $F B_{m-1}$ と今回処理中のピクセルの色データ $P I X_m$ とを、それぞれソース混合計数 $S R_m$ とディスティネーション混合計数 $D E_m$ とでブレンディングすることにある。即ち、演算式で表すと、今回の処理で求められてフレームバッファ内に格納される色データ $F B_m$ は、

【0069】上記のブレンディング演算式において、ソース混合計数 $S R_m$ とディスティネーション混合計数 $D E_m$

E_m とは、次の11種類のうちから適宜選択される。

【0070】 $0, FB_{n-1}, PIX_n, PIX_n * \alpha, FB_{n-1} * \alpha, 1 - FB_{n-1}, 1 - PIX_n, 1 - PIX_n * \alpha, 1 - FB_{n-1} * \alpha, (FB_{n-1} * \alpha, 1 - PIX_n * \alpha)$ のうちいずれか小さい方

上記の式(8)から理解される通り、一般的なブレンディング処理は、描画前のフレームバッファの色データを必要とし、また演算結果も描画順序に依存している。従って、一般的なコンピュータ・グラフィックにおいては、ブレンディング処理の必要なポリゴンは奥側より順

$$FB_m = PIX_m * \alpha_m + FB_{m-1} * (1 - \alpha_m) \quad (9)$$

即ち、上記式(9)は、一般式(8)において、ディスティネーション混合係数 $DE_m = \alpha_m$ で、ソース混合係数 $SR_m = (1 - \alpha_m)$ にした場合に対応する。

【0072】図8は、補助フィルタ処理の具体例を示す図である。汎用的なグラフィックライブラリには、ブレンド処理として補助フィルタ処理がある。この処理は、図7の半透明ポリゴンの混合ブレンディング処理と異なり、奥側のポリゴン PG_{m-1} の色データがその手前側のポリゴン PG_m の混合係数 α_m にかかわらず、そのまま減衰することなく残される処理である。

【0073】例えば、手前側のポリゴン PG_m が炎に対応するポリゴンであり、奥側のポリゴン PG_{m-1} が所定の不透明ポリゴンの場合を想定する。この場合は、視点からは炎のポリゴン PG_m を通して不透明ポリゴン PG_{m-1} を見ていることになる。その場合は、炎ポリゴン PG_m の色は加算されるが、奥側のポリゴン PG_{m-1} の色データが減衰することはない。

【0074】その場合は、炎ポリゴン PG_m の色データ PIX_m に対してディスティネーション混合係数 $DE_m = \alpha_m$ 、ソース混合係数 $SR_m = 1$ にすることにより、上記の一般式(8)を利用することができる。その結果、補助

$$FB_m = PIX_m * 0 + FB_{m-1} * PIX_m \quad (10)$$

となる。この演算式は、一般式(8)に対して、カラーフィルタポリゴン PG_m の色抽出データ PIX_m に対してディスティネーション混合係数 $DE_m = 0$ 、前回求めた色データ FB_{m-1} に対してソース混合係数 $SR_m = PIX_m$ することにより得られる。

【0078】以上の通り、半透明ポリゴンの混合ブレンディング処理以外に、補色フィルタ処理やカラーフィルタ処理などの様々なブレンディング処理は、一般式(8)に、適宜選択したディスティネーション混合係数 DE_m とソース混合係数 SR_m と適用して演算処理することができる。通常のコンピュータ・グラフィック処理装置は、上記の一般式(8)に適用できるように構成され、その結果、ディスティネーション混合係数 DE_m とソース混合係数 SR_m とを適宜選択して行うことにより、様々なブレンディング処理が可能になる。

【0079】そこで、上記の一般的なブレンディング処理に対して、視点側から(手前側)ポリゴンを順番に処

に描画するのが通例である。

【0071】図7は、半透明ポリゴンの混合ブレンディング処理を示す図である。手前側の半透明ポリゴン PG_m と、奥側のポリゴン PG_{m-1} が一部重なっていて、手前側の半透明ポリゴン PG_m の不透明度が α の場合である。この場合は、従来例で説明した通り、奥側のポリゴン PG_{m-1} の色データ FB_{m-1} が先に求められ、それに手前側の半透明ポリゴン PG_m の色データが、不透明度 α_m と透明度 $(1 - \alpha_m)$ によって、混合ブレンディング処理される。即ち、その演算式は、図示される通り、

$$FB_m = PIX_m * \alpha_m + FB_{m-1} * (1 - \alpha_m) \quad (11)$$

フィルタ処理の演算式は、図示される通り、

$$FB_m = PIX_m * \alpha_m + FB_{m-1} * (1 - \alpha_m) \quad (12)$$

となる。即ち、前回の処理で求めたポリゴン PG_{m-1} の色データ FB_{m-1} をそのまま残し、今回処理中のポリゴン PG_m の色データ PIX_m に混合係数 α_m 分だけ加算する処理である。

【0075】この補助フィルタ処理は、上記の炎ポリゴンの例以外に、例えば赤い光源からの光がポリゴン PG_{m-1} に照射されている場合も、同様の処理が適用される。

【0076】図9は、更に、カラーフィルタ処理を示す図である。カラーフィルタ処理では、例えば、背景にあるポリゴン PG_{m-1} の特定の色成分だけを抽出するため、手前側にそれ自身は色成分を持たないカラーフィルタポリゴン PG_m を配置させる。そして、このブレンディング処理演算では、特定の色成分だけを抽出するために、背景のポリゴン PG_{m-1} の色データ FB_{m-1} にカラーフィルタポリゴン PG_m の色抽出データ PIX_m を乗算する。

【0077】即ち、その演算式は、図示される通り、

$$FB_m = PIX_m * \alpha_m + FB_{m-1} * (1 - \alpha_m) \quad (13)$$

理することができる方法、処理装置について、以下に説明する。

【0080】図1の場合と同様に、4枚のポリゴンが手前側から重なっていると仮定する。その場合、一般式(8)に対して、 $m = 1 \sim 4$ により、次の4つの演算式が導かれる。

【0081】

$$FB_1 = PIX_1 * DE_1 + FB_0 * SR_1 \quad (21)$$

$$FB_2 = PIX_2 * DE_2 + FB_1 * SR_2 \quad (22)$$

$$FB_3 = PIX_3 * DE_3 + FB_2 * SR_3 \quad (23)$$

$$FB_4 = PIX_4 * DE_4 + FB_3 * SR_4 \quad (24)$$

ここで、留意すべき点は、上記ブレンディング処理は、奥側のポリゴンを先に処理してその色データがフレームバッファに記録されているので、 m が大きいほど手前のポリゴンを意味し、 m が小さいほど奥側のポリゴンを意味することである。

【0082】そこで、上記の(21)～(24)について

て、式(5) (6) (7)で行ったように、最も手前のポリゴンの色データを F_{B_1} から D_1 に、最も奥側のポリゴンの色データを F_{B_n} から D_n にそれぞれ置き換え、処理中のピクセルの色データ PIX_i をテクスチャーデータ(色データ) T_i に置き換えると、次の通りである。つまり、 $D_i = F_{B_1} \sim D_1 = F_{B_n}$ とし、 $PIX_i = T_1 \sim PIX_n = T_n$

$$X_n = SR_n * X_{n-1} = SR_n * SR_{n-1} * SR_{n-2} * \dots * SR_1 \quad (35)$$

とすると、式(34)は、次の通り展開される。

【0083】 $D_1 = T_1 * DE_1 * X_1 + T_2 * DE_2 * X_1 + T_3$
 $D_n = D_{n-1} + T_n * DE_n * X_{n-1}$

と一般化することができる。この場合上記の置き換えにより、 n が小さいほど手前のポリゴンを意味しているので、より手前のポリゴンから順番にブレンディング処理を行うことができることを意味している。即ち、より手前に位置するポリゴンの処理結果 D_{n-1} を利用して、次の奥側に位置するポリゴンのデータ D_n を求めることができるのである。そして、上記の式(35)と(36)において、 $SR_n = (1 - \alpha_n)$ 、 $DE_n = \alpha_n$ と置き換えると、半透明ポリゴンの処理で示した式(6) (7)と同じでになることが理解される。

【0084】上記の式(35)によれば、ワークデータ X_n は、ソース混合係数 SR_n の累積値である。また、上記の式(36)によれば、視点から n 枚目のポリゴンを処理した結果の色データ D_n は、 n 枚目のポリゴンのテクスチャデータ(色データ) T_n を、ディスティネーション混合係数 DE_n と、手前に位置する $n - 1$ 枚のポリゴンのソース混合係数 $SR_1 \sim SR_{n-1}$ の累積値 X_{n-1} とに応じて、 $n - 1$ 枚目までの画像データ D_{n-1} に加えるレンダリング処理により求められる。しかも、ワークデータ X_n が所定の最低値よりも小さくなると、その後のポリゴンの処理を省略しても表示される色データに大きな影響を与えないことは、半透明ポリゴンのブレンド処理と同じである。

【0085】図10は、一般的なグラフィック・ライブラリでのブレンディング処理に利用される汎用レンダラの構成図である。図5のレンダラの構成を汎用化したものであり、図4のレンダラ34に適用される。図10の各構成の引用番号は、図5の対応する構成と同じ引用番号を与えた。

【0086】図10のレンダラは、図5のレンダラと異なり、ディスティネーション混合係数生成部348と、ソース混合係数生成部349とが設けられ、更に、乗算器は、 X_n 乗算器344Xと、 D_n 乗算器344Dとが設けられる。また、制御ブロック341は、描画コマンド36として、テクスチャーベクタ(T_x, T_y)と、不透明度 α と、混合係数生成情報 iDE, iSR を受信する。それ以外の構成は、図5の場合と同じである。

半透明ポリゴンの混合ブレンディング処理 : $DE_n = \alpha_n, SR_n = 1 - \alpha_n$
 補助フィルタ処理 : $DE_n = \alpha_n, SR_n = 1$
 カラーフィルタ処理 : $DE_n = 0, SR_n = T_n$

T_i すると、

$$D_1 = T_1 * DE_1 + D_1 * SR_1 \quad (31)$$

$$D_2 = T_2 * DE_2 + D_2 * SR_2 \quad (32)$$

$$D_3 = T_3 * DE_3 + D_3 * SR_3 \quad (33)$$

$$D_n = T_n * DE_n + D_n * SR_n \quad (34)$$

そこで、ワークデータ X_n (但し $X_0 = 1$) を、

$$* DE_1 * X_2 + T_1 * DE_1 * X_1 \quad (35)$$

であるから、各 $T_n * DE_n * X_{n-1}$ に注目すると、

$$(36)$$

【0087】混合係数生成情報 iDE, iSR は、一般化されたブレンド処理の中から選択される所定の処理に対応して、ディスティネーション混合係数 DE とソース混合係数 SR をどれにすべきかを指示する情報を含む。従って、この混合係数生成情報 iDE, iSR は、それぞれの混合係数生成部348, 349に与えられ、それらの生成部348, 349が、ディスティネーション混合係数 DE とソース混合係数 SR を生成して、 X_n 乗算器344Xと D_n 乗算器344Dとに供給する。

20 【0088】制御ブロック341は、描画コマンドに含まれるテクスチャーベクタ(T_x, T_y)に従って、テクスチャーバッファ44から対応するテクスチャデータを読み出し、テクスチャデータ(色データ) T_n として、 D_n 乗算器344Dに供給する。また、制御ブロックは、Xバッファ342とフレームバッファであるDバッファ343に対して、処理中のピクセルに対応するアドレスへのアクセスを指示し、それらの読み出されたデータ X_{n-1} と D_{n-1} は、それぞれ X_n 乗算器344Xと、加算器347に供給される。

30 【0089】上記の汎用レンダラにより、視点に対してより手前側のポリゴンから順番にブレンディング処理が行われる。そして、 n 番目のポリゴンの処理において、 X_n 乗算器344Xは、前回処理してXバッファ342内に記録していたワークデータ X_{n-1} と今回のソース混合係数 SR_n とから、ワークデータ X_n を求める。更に、 D_n 乗算器344Dと加算器347とは、前回処理してXバッファ342内に記録していたワークデータ X_{n-1} 、今回のディスティネーション混合係数 DE_n 、及び前回処理してDバッファ343内に記録していた色データ D_{n-1} とから、今回の色データ $D_n = D_{n-1} + T_n * DE_n * X_{n-1}$ を求める。それぞれ求められたワークデータ X_n と色データ D_n とは、それぞれXバッファ342とDバッファ(フレームバッファ)343に記録される。

40 【0090】図10の汎用レンダラでは、次の様に混合係数を選択することで、上記した3つの処理を行うことができる。

【0091】

上記の例以外にも、種々の混合係数を選択することができる。例えば、 $0, T_n * \alpha_n, 1, 1 - T_n, 1 - T_n * \alpha_n$ 等を適宜選択することができる。但し、データ FB_{n-1} 、即ち、後で処理して求められる色データ D_{n+1} を含む混合係数を選択することはできない。

【0092】汎用レンダラの場合は、好ましくは、図10に設けられた構成をRGBの色毎に有する。但し、RGB各色成分に同一の混合係数、例えば $0, \alpha, 1, 1 - \alpha$ のみを使用する場合は、共通のXバッファ342と各色成分のDバッファ343で構成することができる。また、色成分で異なる混合係数、例えば $T_n, 1 - T_n$ を含む場合は、各色成分毎のXバッファ342とDバッファ343で構成する必要になる。

【0093】そして、上記の汎用レンダラでも、n番目のポリゴンに対するブレンディング処理において、ソース混合係数 SR_n の累積値が所定の最小値 X_{n+1} より小さい場合は、n番目以降のポリゴンに対する処理を省略することができ、全体のブレンディング処理の効率を上げることができる。

【0094】上記の汎用レンダラの如きハードウェアでなく、ソフトウェアを利用して、上記の処理を行うことも可能である。その場合は、汎用のパーソナルコンピュータやワークステーションに、上記のブレンド処理プログラムを実行させることにより実現される。

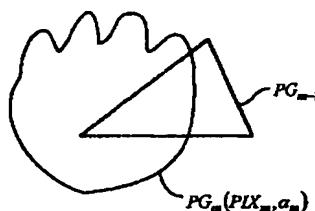
【0095】

【発明の効果】以上説明した通り、本発明によれば、半透明ポリゴンの画像処理において、表示画面の奥行き方向について手前側のポリゴンから順に処理することができ、処理すべき全てのポリゴンを処理しない場合でも、再生成される画像のスムーズな動きに大きな影響を与えることはないので、高速処理を可能にする。

【0096】また、本発明によれば、不透明ポリゴンの処理と区別することなく半透明ポリゴンの処理を行っても、不透明ポリゴンの処理効率を高くすることができ、また、区別する必要がないのでその分処理の負担を軽減

【図8】

補助フィルタ処理の具体例



$$FB_m = PIX_m * \alpha_m + FB_{m-1} * 1$$

$$(DE_m) \quad (SR_m)$$

することができる。

【0097】更に、本発明によれば、一般的なブレンディング処理を行うことができる汎用レンダラを提供することができ、そのディスティネーション混合係数やソース係数を適宜選択することで、種々のブレンディング処理を、視点に対して手前側のポリゴンから順番に処理することができる。

【図面の簡単な説明】

【図1】半透明ポリゴンの重なり例を説明する図である。

【図2】本発明及び従来例のレンダリング処理を、図1の例に適用した時の演算式を示す図表である。

【図3】図2に示したレンダリング処理の演算を具体例に適用した例を示す図表である。

【図4】本発明の画像処理装置にかかるゲーム装置の概略構成例を示す図である。

【図5】本発明の実施の形態例におけるレンダラの構成図である。

【図6】本実施の形態例の画像処理のフローチャート図である。

【図7】半透明ポリゴンの混合ブレンディング処理を示す図である。

【図8】補助フィルタ処理の具体例を示す図である。

【図9】カラーフィルタ処理を示す図である。

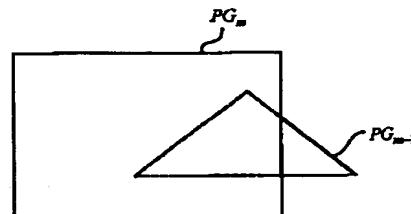
【図10】一般的なグラフィック・ライブラリでのブレンディング処理に利用される汎用レンダラの構成図である。

【符号の説明】

34	レンダラ、レンダリング処理部
342	Xバッファ、ワークデータバッファ
343	Dバッファ、画像データバッファ
T_n	色データ、テクスチャデータ
D_n	画像データ
α_n	不透明度データ
$P_1 \sim P_4$	ポリゴン

【図9】

カラーフィルタ処理の具体例

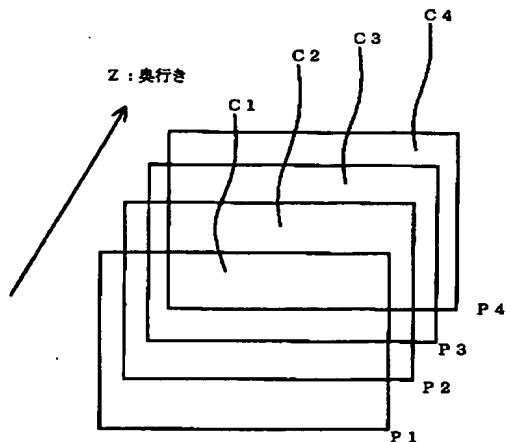


$$FB_m = PIX_m * 0 + FB_{m-1} * PIX_m$$

$$(DE_m) \quad (SR_m)$$

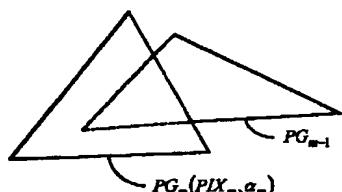
【図 1】

半透明ポリゴンの重なり



【図 7】

半透明ポリゴンのブレンディング処理



$$FB_m = PIX_m * \alpha_m + FB_{m-1} * (1 - \alpha_m)$$

$$(DE_m) \quad (SR_m)$$

【図 2】

(1)

本発明

	n	Xn	Dn
初期値	0	1	0
	1	$X_0(1 - \alpha_1)$	$D_0 + T_1 \alpha_1 X_0$
	2	$X_1(1 - \alpha_2)$	$D_1 + T_2 \alpha_2 X_1$
	3	$X_2(1 - \alpha_3)$	$D_2 + T_3 \alpha_3 X_2$
結果	4	$X_3(1 - \alpha_4)$	$D_3 + T_4 \alpha_4 X_3$

(2)

従来例

	n	Cn
初期値	-	C_5
	4	$C_4 = \alpha_4 T_4 + (1 - \alpha_4) C_5$
	3	$C_3 = \alpha_3 T_3 + (1 - \alpha_3) C_4$
	2	$C_2 = \alpha_2 T_2 + (1 - \alpha_2) C_3$
結果	1	$C_1 = \alpha_1 T_1 + (1 - \alpha_1) C_2$

【図 3】

(1) 本発明

ポリゴンデータ

	α (比)	R(赤)	G(緑)	B(青)
手前から順に	0.500	1.000	0.000	0.000
	0.500	0.000	1.000	0.000
	0.500	1.000	0.000	0.000
背景	1.000	0.000	0.000	0.000

レンダリング結果

	n	Xn	Dn(R)	Dn(G)	Dn(B)
初期値	0	1.000	0.000	0.000	0.000
	1	0.500	0.500	0.000	0.000
	2	0.250	0.500	0.000	0.250
	3	0.125	0.625	0.125	0.250
結果	4	0.000	0.625	0.125	0.250

(2) 従来例

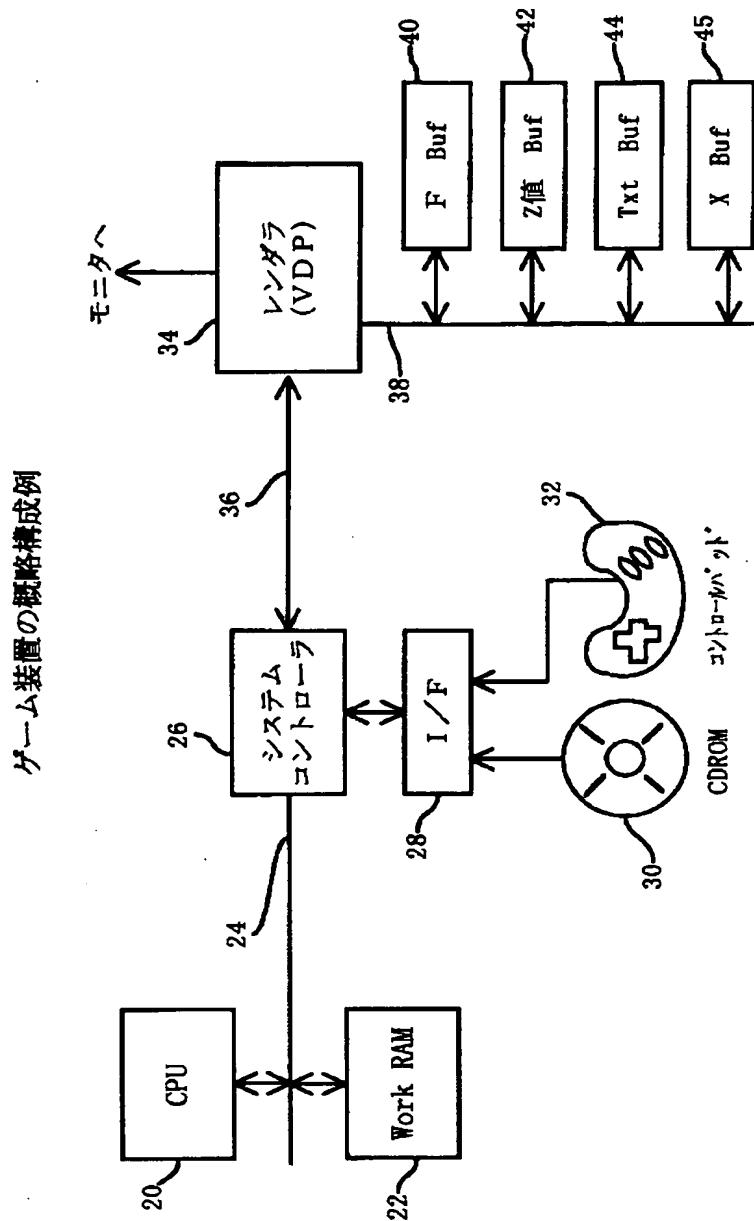
ポリゴンデータ

	α (比)	R(赤)	G(緑)	B(青)
背景	1.000	0.000	0.000	0.000
奥から順に	0.500	1.000	1.000	0.000
	0.500	0.000	0.000	1.000
	0.500	1.000	0.000	0.000

レンダリング結果

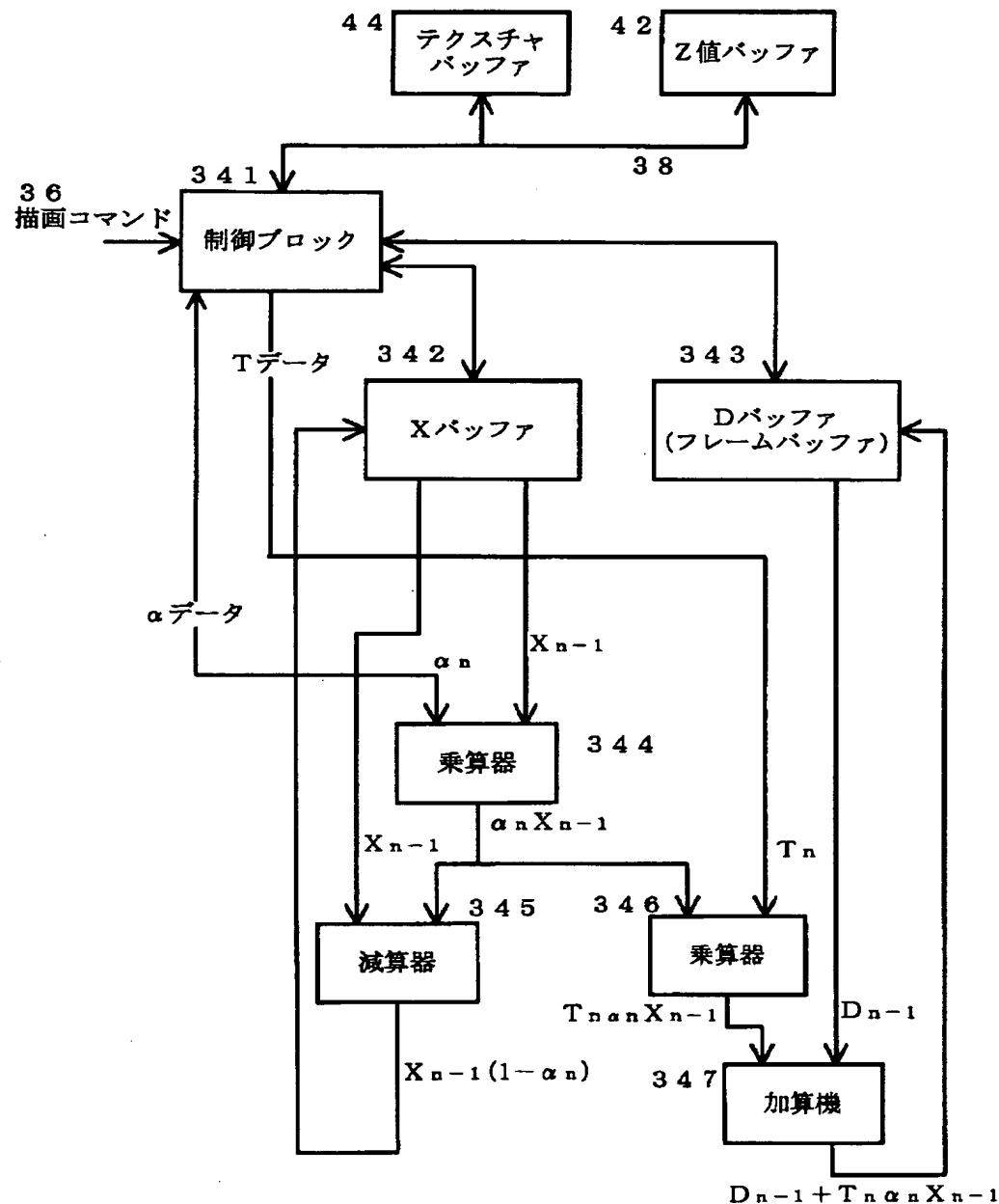
	n	Cn(R)	Cn(G)	Cn(B)
初期値	*	0.000	0.000	0.000
	4	0.000	0.000	0.000
	3	0.500	0.500	0.000
	2	0.250	0.250	0.500
結果	1	0.625	0.125	0.250

【図 4】



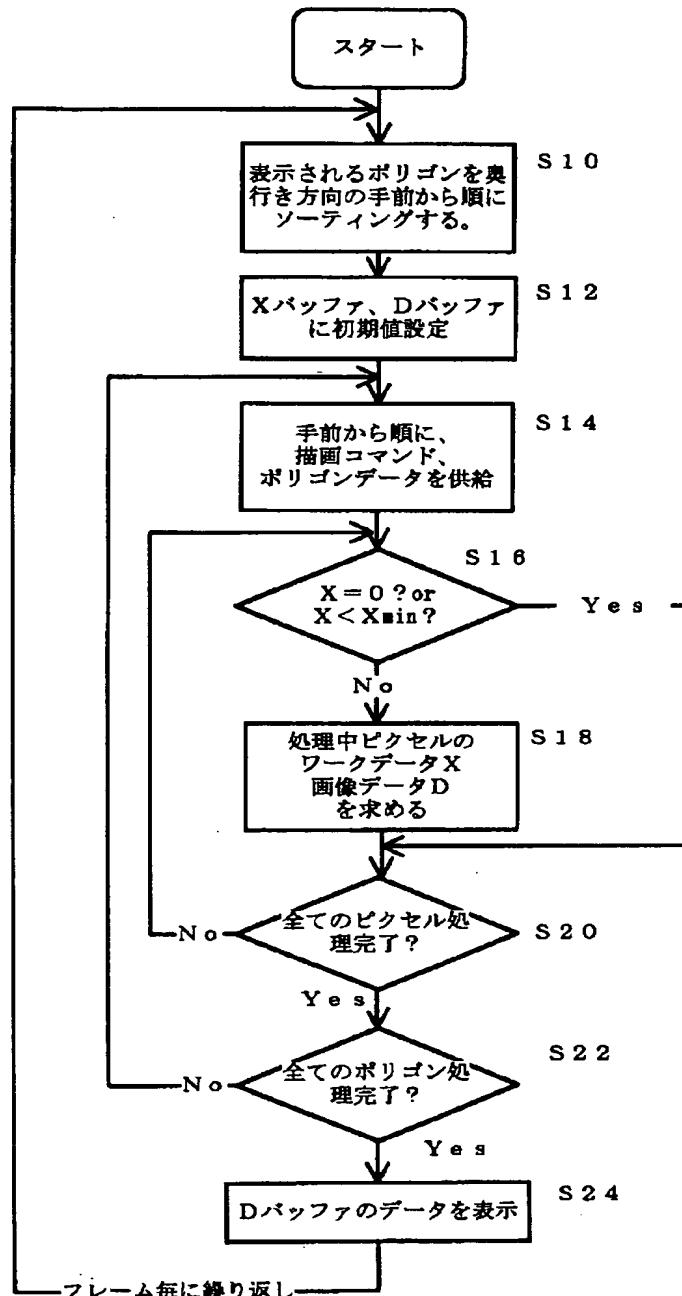
【図5】

レンダラの構成図



【図6】

画像処理のフローチャート



【図 10】

汎用レンダラの構成図

